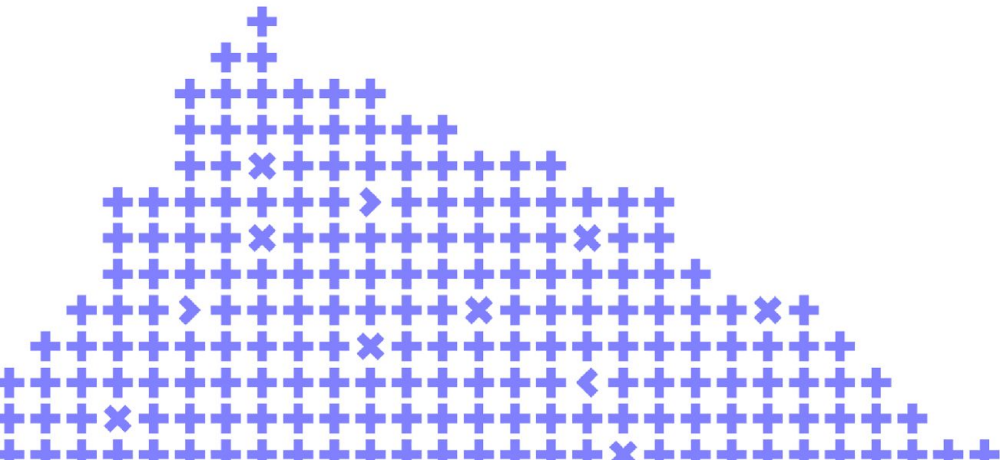


# NoSQL and transactions:

getting the numbers out



Co-organizer

Yandex

# Konstantin Osipov



Director, Software Engineering, ScyllaDB

- + LWT/Paxos/Raft at ScyllaDB
- + co-founder picodata.io
- + ex-CTO Tarantool

# TRANSACTIONAL NOSQL MARKET



# Goals of the test

- ACID properties
- scalability
- price/performance: is it better than RDBMS?
- <http://github.com/picodata/stroppy>



[www.stroppy.io](http://www.stroppy.io)

# Test subjects

- PostgreSQL
- FoundationDB
- CockroachDB
- MongoDB
- YDB?

# Applying the banking test one more time

- load account data
- run money transfers
  - nemesis,
  - transaction distribution: Uniform, Zipfian
- check total balance, check no transaction is lost

# PostgreSQL illustration: accounts and transfers

```
CREATE TABLE IF NOT EXISTS account (  
    bic TEXT, -- bank identifier code  
    ban TEXT, -- bank account number within the bank  
    balance DECIMAL, -- account balance  
    PRIMARY KEY(bic, ban)  
);  
CREATE TABLE IF NOT EXISTS checksum (  
    name TEXT PRIMARY KEY,  
    amount DECIMAL  
);
```

# PostgreSQL illustration - accounts and transfers

**BEGIN**

**UPDATE** account **SET** balance = balance - \$1

**WHERE** bic = \$2 and ban = \$3

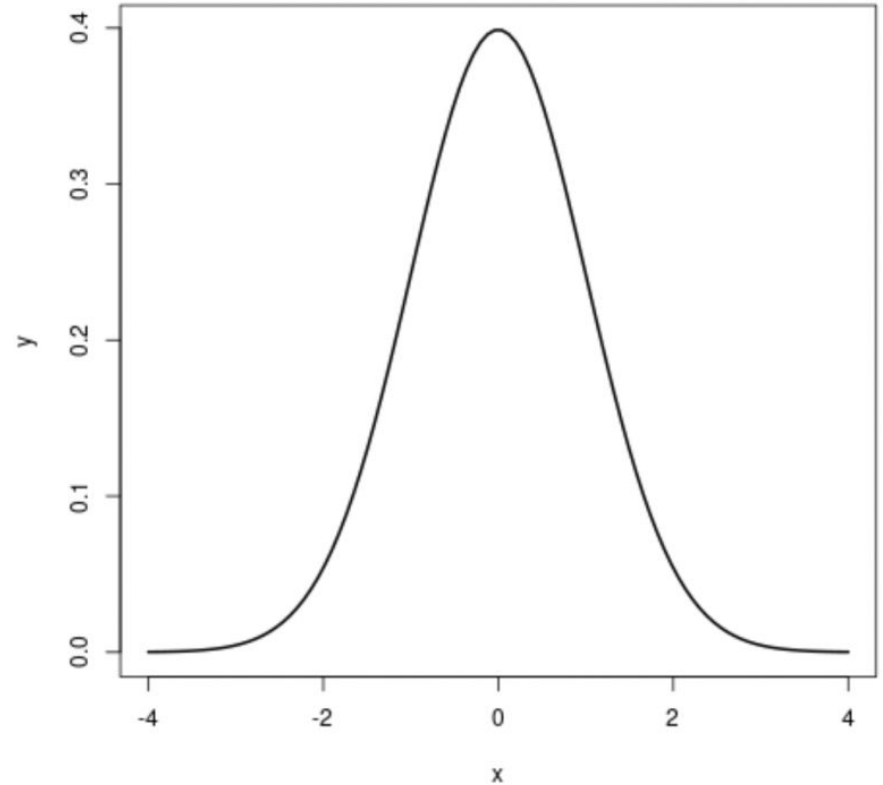
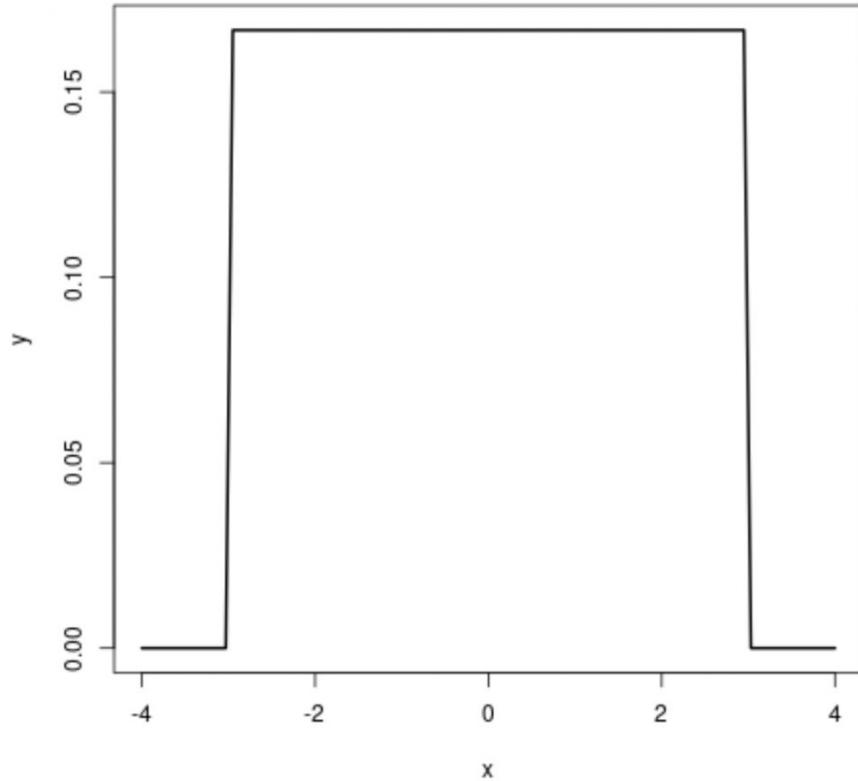
**UPDATE** account **SET** balance = balance + \$1

**WHERE** bic = \$2 and ban = \$3

**COMMIT**



# Transfer account number distribution



# Hardware

Oracle Cloud:

*1Gb of disk/network  
bandwidth per core*

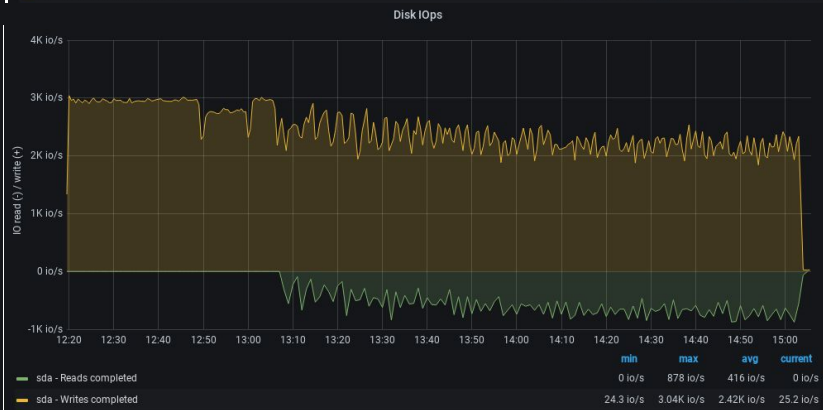
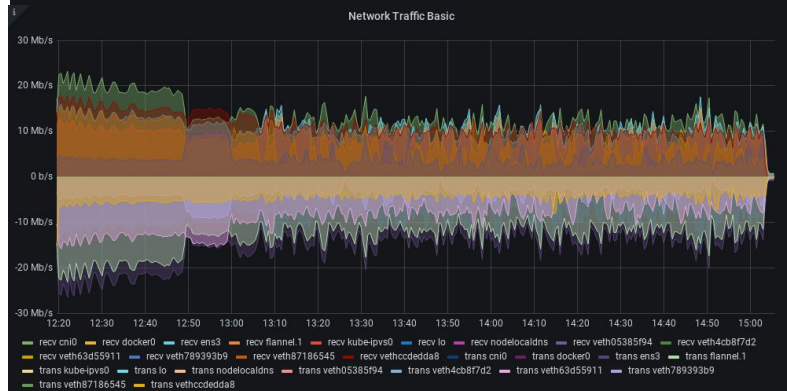
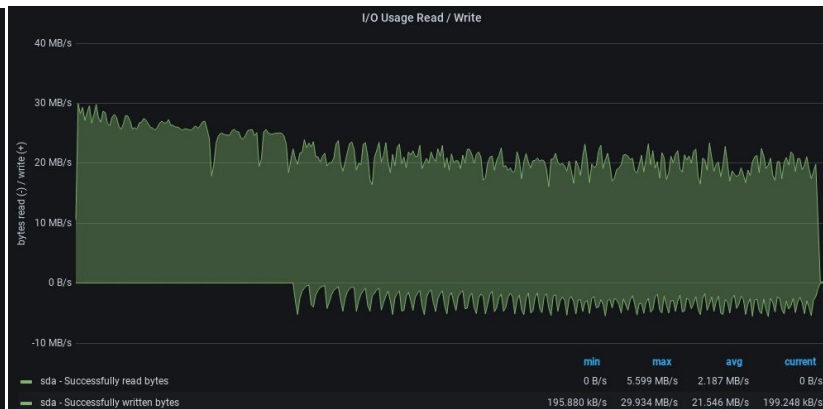
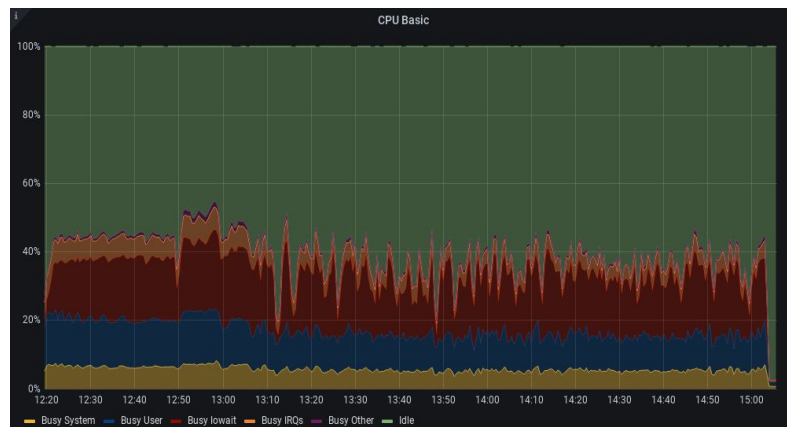
Yandex Cloud:

*Intel Xeon® Processor  
E5-2660 v4, 4 Gb  
RAM, Network-SSD 15  
Gb*

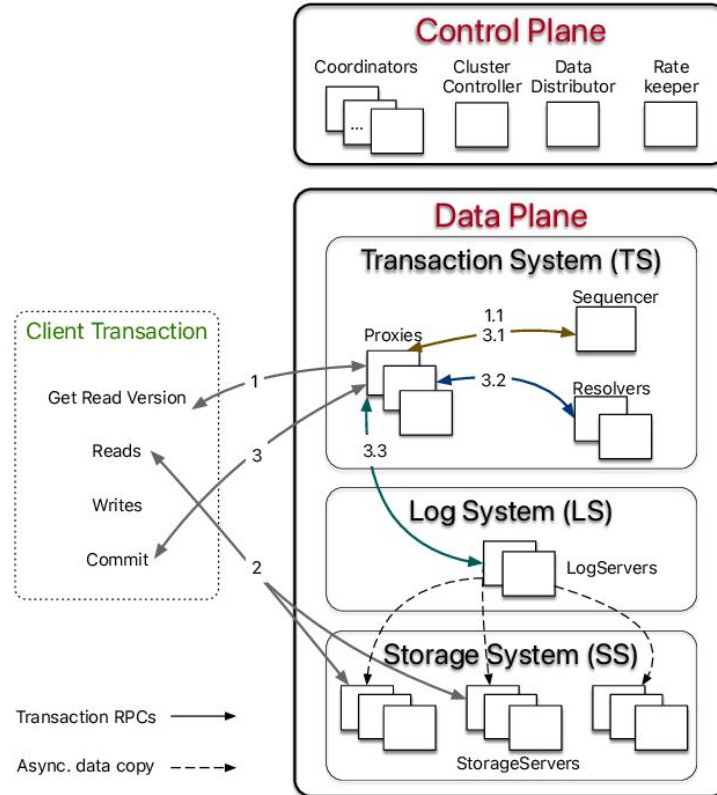
# PostgreSQL results

VCPU/ node	RAM/ node, GB	HDD/ Node, GB	Clients	Accounts, millions	Transfers, millions	TPS
3	30	100	128	10	100	2059
10	160	1000	256	100	100	<b>5915</b>

# PostgreSQL limits



# FoundationDB architecture

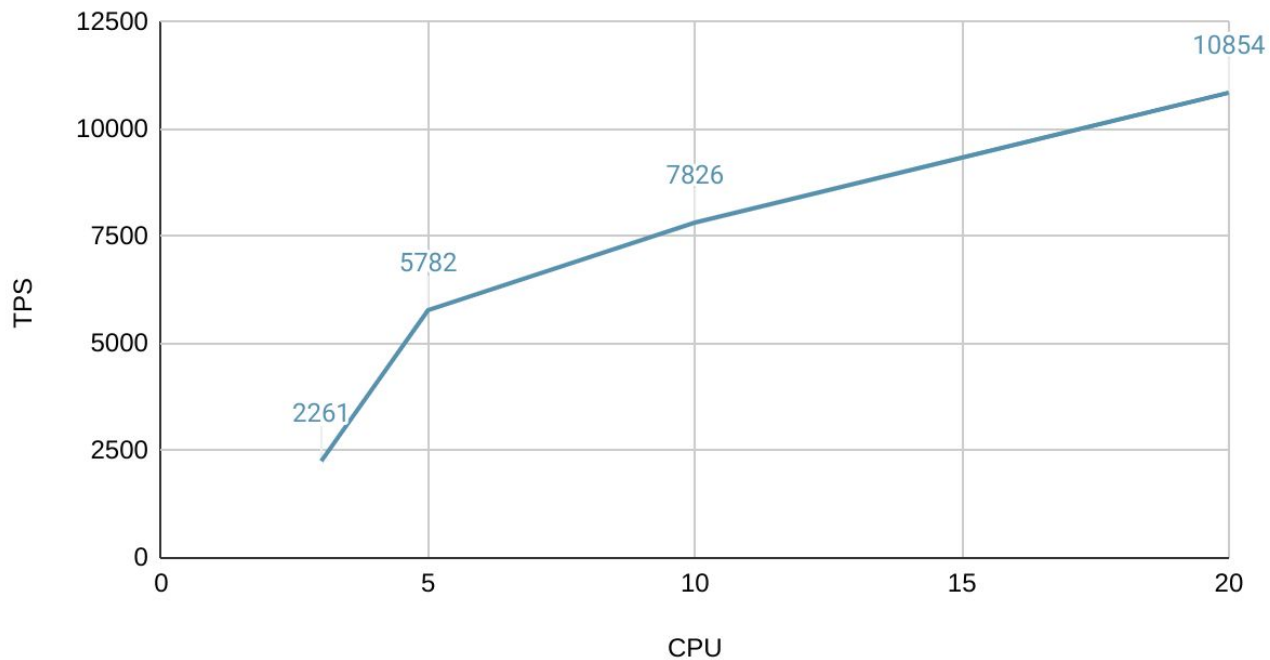


# FoundationDB results

Nodes	VCPU/ node	RAM/ node, GB	HDD/ Node, GB	Clients	Account s, millions	Transfer s, millions	TPS
3	1	8	100	16	10	100	2263
3	2	8	100	16	10	100	2189
5	1	8	100	512	10	100	7631
5	1	16	100	512	100	100	5782
20	1	16	100	512	100	100	10854

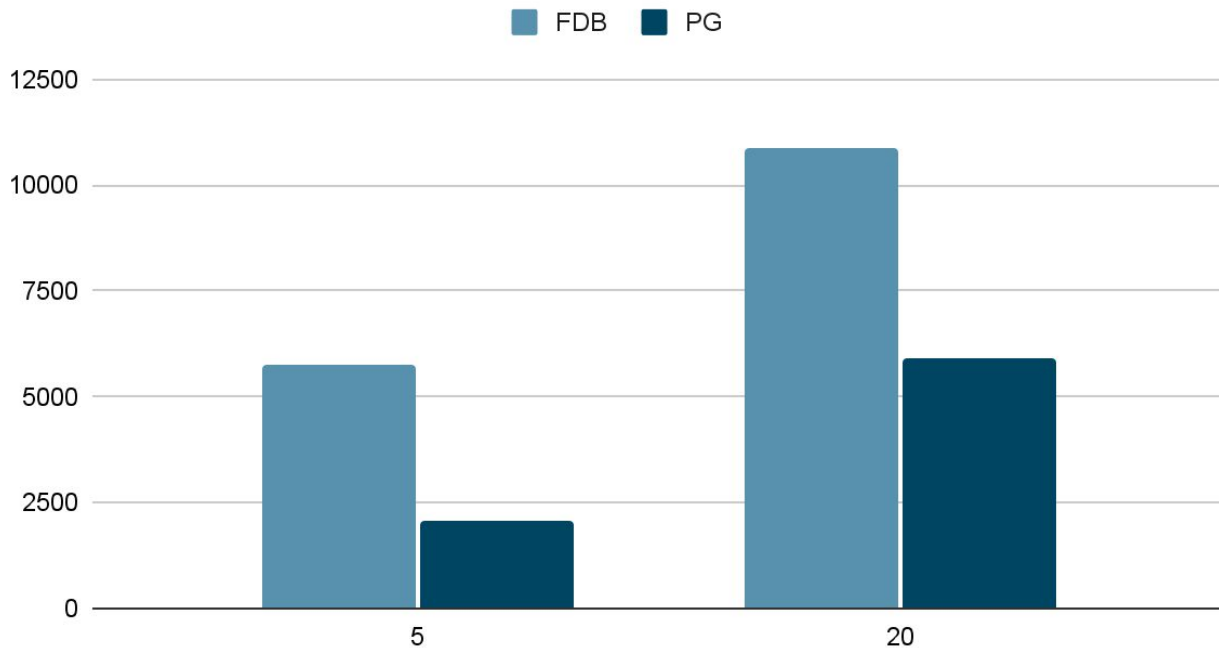
# FoundationDB results

TPS относительно CPU



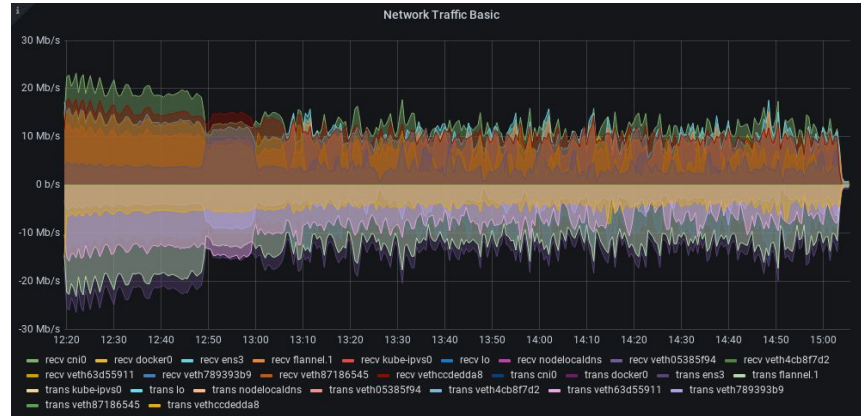
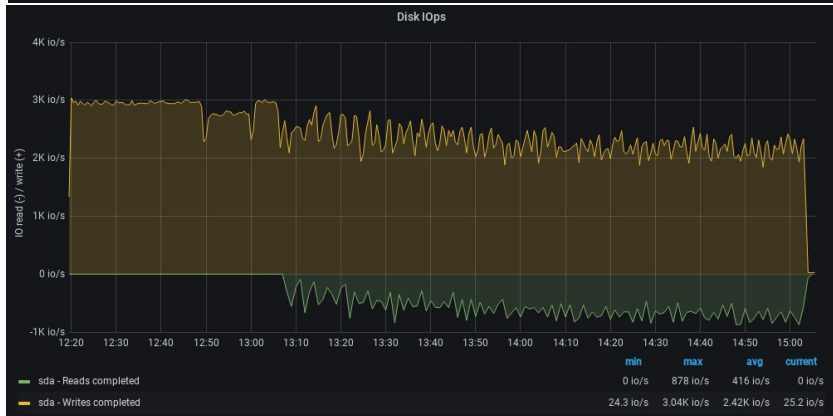
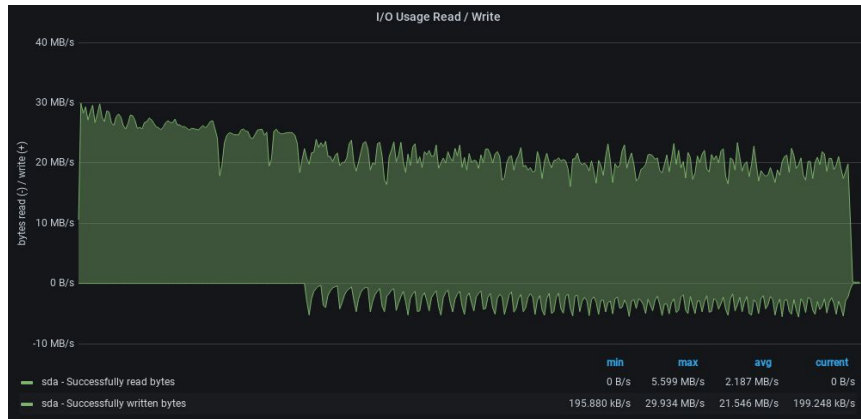
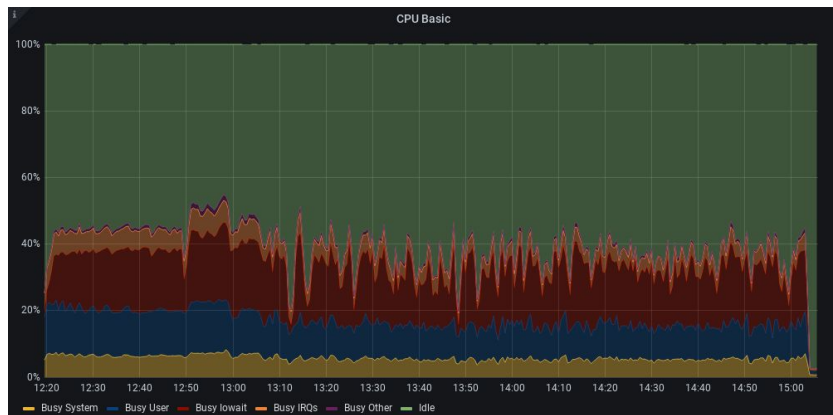
# FoundationDB vs PostgreSQL

TPS per core

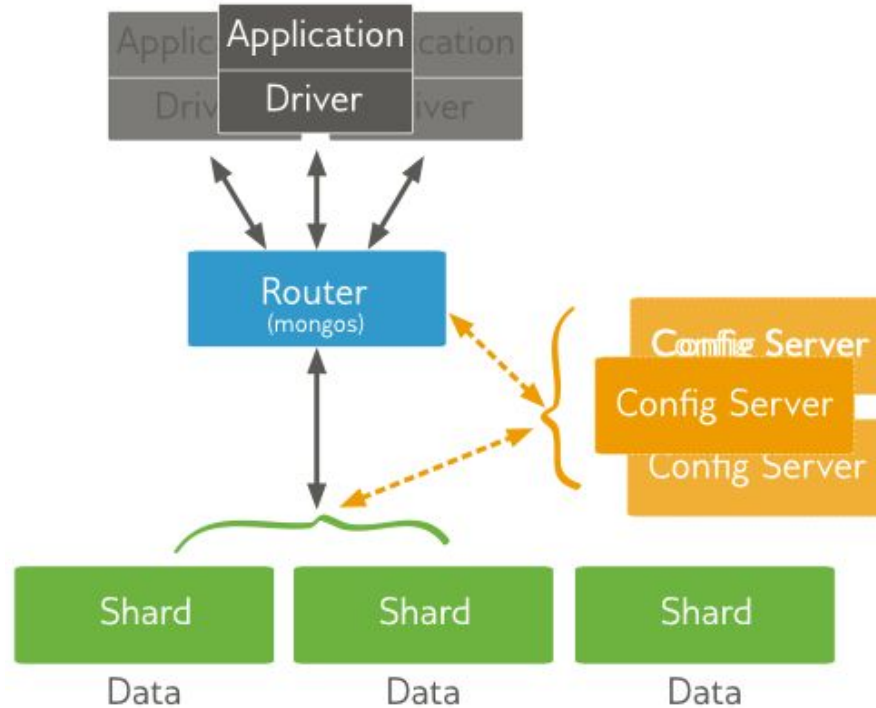




# FoundationDB bottlenecks



# MongoDB architecture



# MongoDB - vertical scalability

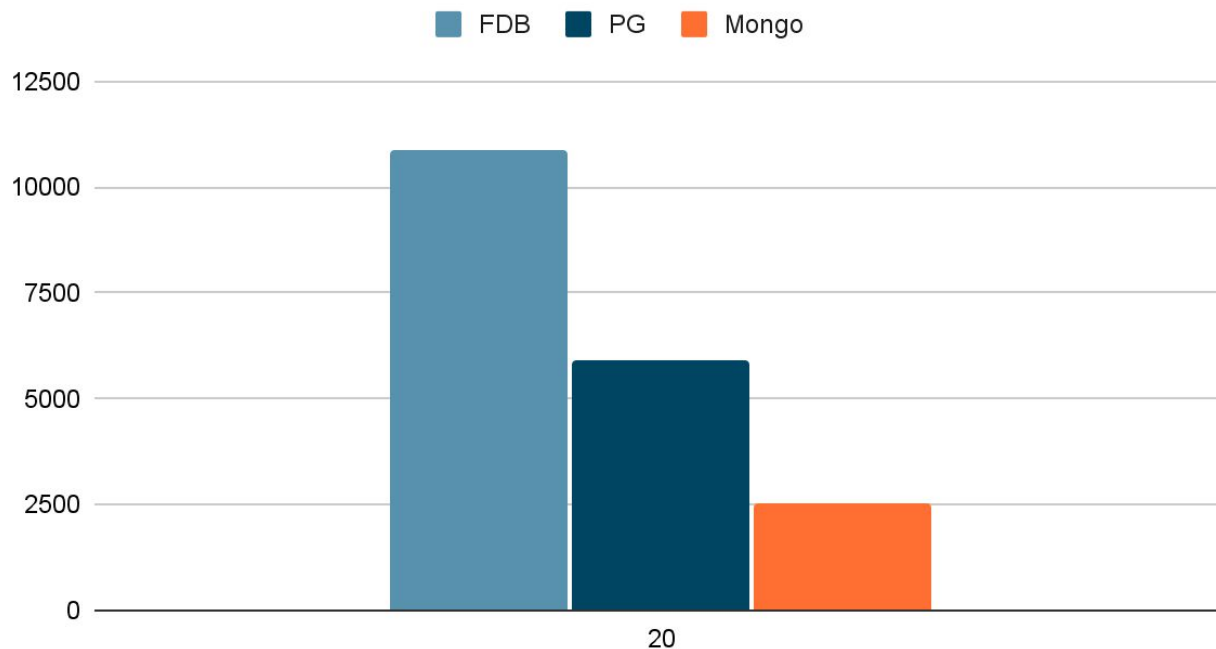
<b>VCPU/ node</b>	<b>RAM/ node, GB</b>	<b>HDD/ Node, GB</b>	<b>Clients</b>	<b>Accounts, millions</b>	<b>Transfers, millions</b>	<b>TPS</b>
2	8	100	16	10	10	720
4	8	100	128	10	10	1843
6	16	100	128	100	100	2761
12	40	1000	128	1000	100	3272

# MongoDB - **hor**izontal scalability

VCPU/ node	Shards	RAM/ node, GB	HDD/ Node, GB	Clients	Account s, millions	Transfer s, millions	TPS
2	2	8	100	32	10	10	427
4	8	8	100	128	100	10	1171
4	8	8	100	128	1000	100	947

# FoundationDB vs PostgreSQL vs MongoDB

TPS per core



# CockroachDB results

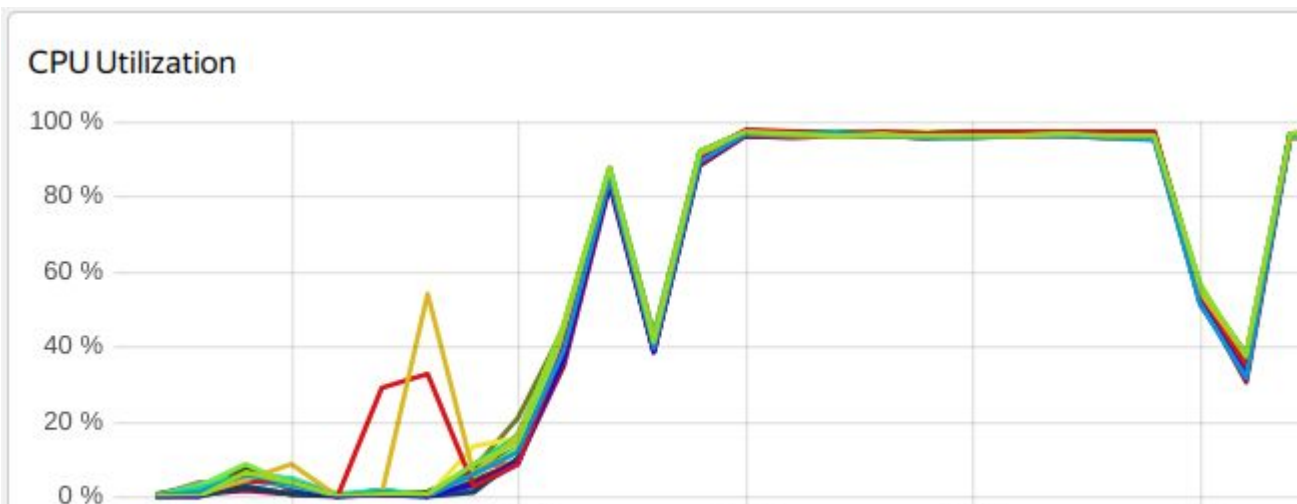
VCPU/ node	Shards	RAM/ node, GB	HDD/ Node, GB	Clients	Account s, millions	Transfer s, millions	TPS
2	3	8	100	32	10	10	2198
1	5	16	100	128	100	10	437
1	20	16	100	128	1000	10	814

## YDB results

VCPU/ node	Shards	RAM/ node, GB	HDD/ Node, GB	Clients	Account s, millions	Transfer s, millions	TPS
16	3	24	100	4000	1	10	15367

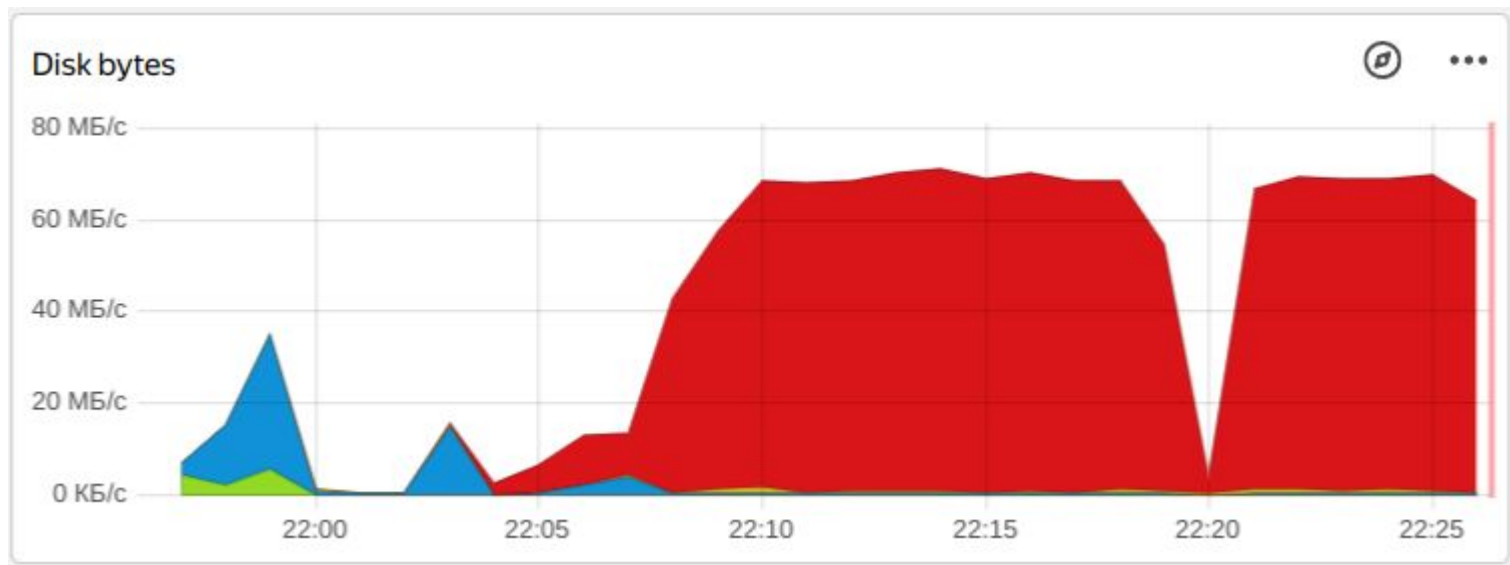
Latency min/99%/max/med: 0.390s/1.129s/3.936s/0.998s

# YDB - load charts - CPU

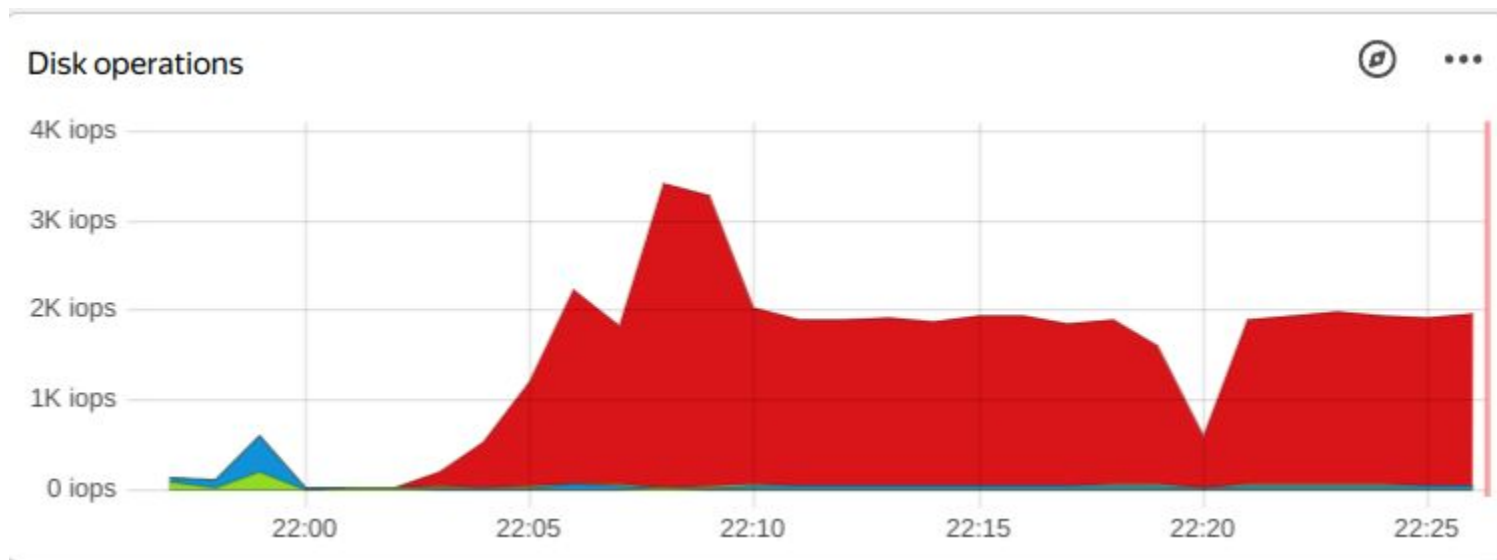




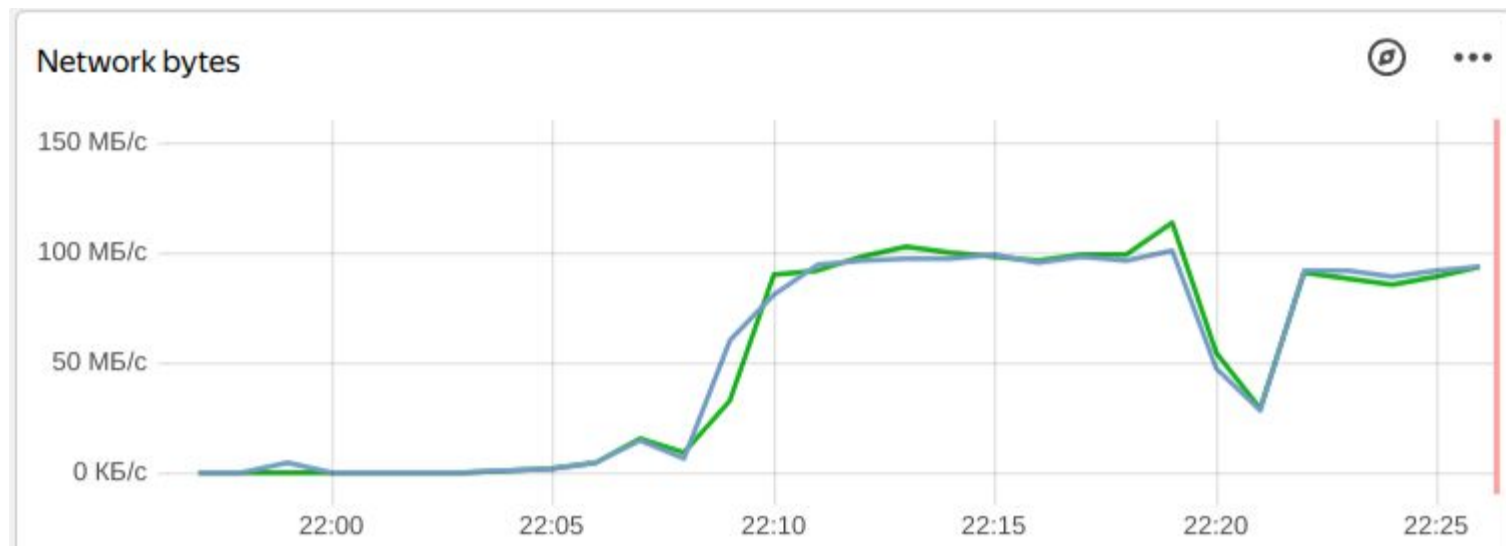
## YDB - load charts - disk



# YDB - load charts - disk



# YDB - load charts - network



# Chaos mesh tests

PostgreSQL -



FoundationDB -



CockroachDB - ?

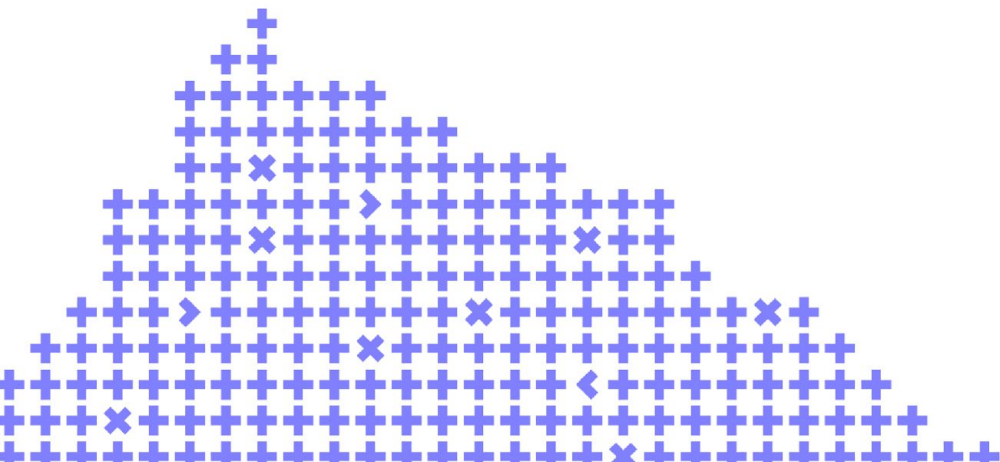
MongoDB -



# Summary

- Scaling transactional workloads horizontally is hard
- FoundationDB has been designed from scratch to support it
- MongoDB is performing the worst
- CockroachDB loses to PostgreSQL on the same hardware
- the Go tool and the results are public at [stroppy.io](https://stroppy.io)

Thank you!



Co-organizer

Yandex